Títol:	**Bicing Stats**
Autor:	Alejandro Carol Campreciós
Data:	14 de Setembre de 2.016
Director:	Xavier Franch Gutiérrez
Departament del Director:	Enginyeria de serveis i sistemes d'informació (ESSI)
Titulació:	Enginyeria Informàtica
Centre:	Facultat d'Informàtica de Barcelona (FIB)
Universitat:	Universitat Politècnica de Catalunya (UPC) Barcelona Tech

DADES DEL PROJECTE

Títol:	**Bicing Stats**
Autor:	Alejandro Carol Campreciós
Data:	14 de Setembre de 2.016
Director:	Xavier Franch Gutiérrez
Departament del Director:	Enginyeria de serveis i sistemes d'informació (ESSI)
Titulació:	Enginyeria Informàtica
Crèdits:	37,5
Centre:	Facultat d'Informàtica de Barcelona (FIB)
Universitat:	Universitat Politècnica de Catalunya (UPC) Barcelona Tech

MEMBRES DEL TRIBUNAL (nom i signatura)

President:	**Xavier Burgués Illa**
Vocal:	**Juan José Navarro Guerrero**
Secretari:	**Xavier Franch Gutiérrez**

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Contingut

1 Introducció

L'any 2007 l'Ajuntament de Barcelona va posar en marxa el servei del Bicing. És un servei d'abonament en el qual l'usuari pot recollir una bicicleta en una estació i circular pedalant fins a una altra estació.

Hi ha aproximadament 420 estacions a la ciutat de Barcelona. Cada estació consisteix en un aparell on l'usuari s'identifica mitjançant la seva targeta. Disposa d'un nombre limitat de places de pàrquing i a cada plaça hi ha lloc per una bicicleta. Quan s'estaciona la bicicleta a la plaça, s'activa un sensor.

Un cop recollida la bicicleta en una estació és obligatori lliurar-la en menys de dues hores a una altra estació. Si l'estació de destí de l'usuari, quan s'arriba, està plena de bicicletes, és l'usuari que ha de buscar una altra estació per desar la bicicleta, que mai es podrà desar al carrer, o s'exposa a una forta penalització econòmica.

Bicing ofereix un servei on s'informa de les places lliures en una determinada estació o zona en el moment que es fa la consulta.

Aquest projecte pretén solucionar la problemàtica amb les que ens trobem els usuaris de Bicing, que no és més que la incertesa de quan trobarem bicicletes o llocs lliures per deixar la bicicleta.

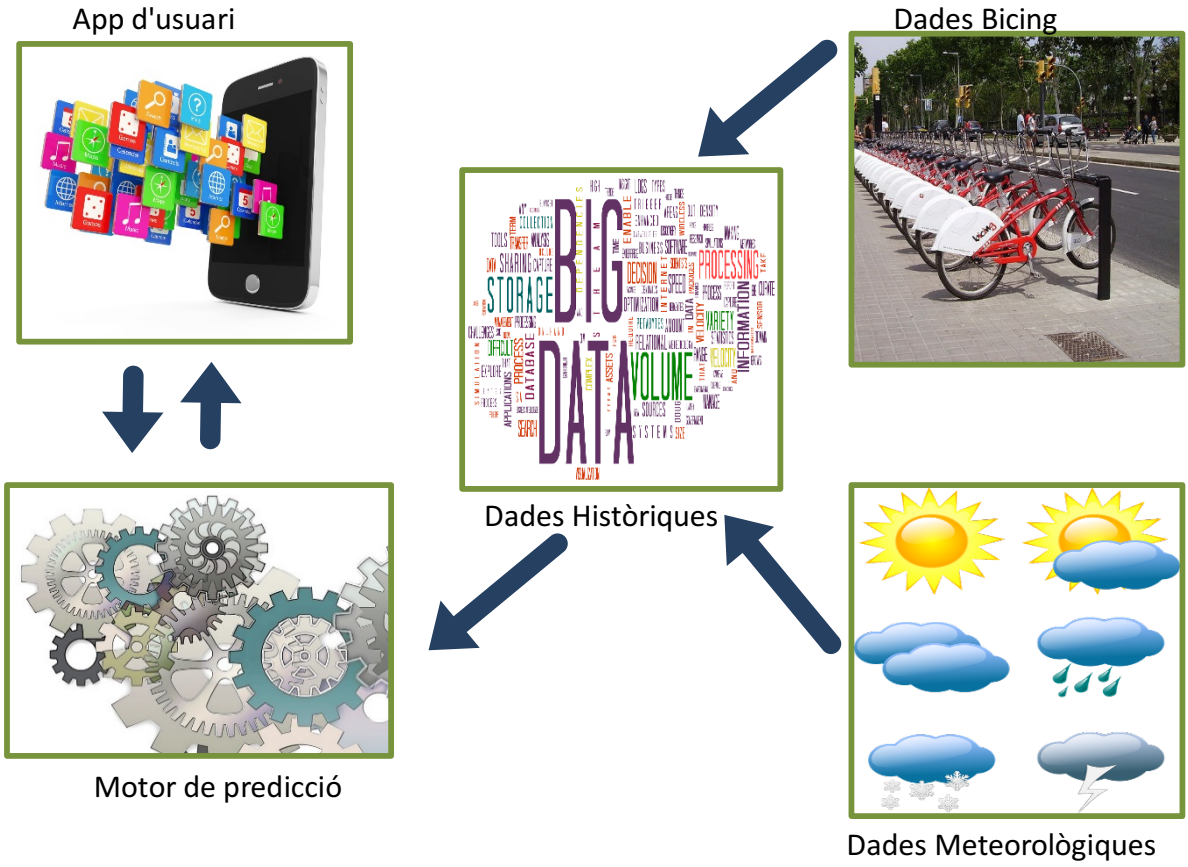
Per solucionar aquesta problemàtica he decidit crear una eina que redueixi aquesta incertesa, **proporcionant prediccions d'ocupació de les estacions a l'usuari** per tal de fer la seva vida una mica més fàcil.

La predicció de disponibilitat es basa en un conjunt de variables, i el pes de cadascun dels factors es calcula segons un procés d'aprenentatge automàtic.

Les variables sobre les quals es construeix l'estimació són:

- L'ocupació actual de l'estació.
- L'historial d'ocupació de l'estació.
- Variables meteorològiques.

Esquema General



1.1 Tipus de Projecte

El present document correspon al Projecte Final de Carrera en modalitat A, projecte realitzat a la UPC.

1.2 Objectiu

L'objectiu del projecte és construir una aplicació que doni una estimació de la probabilitat de trobar una bicicleta lliure en una zona o estació de Bicing.

El projecte tracta de donar rellevància no només a l'algorisme de predicció, sinó també al projecte d'enginyeria de software que envolta aquesta predicció.

L'aplicació serà bàsicament feta per dispositius mòbils, ja que s'espera que els usuaris vulguin obtenir aquestes dades durant diferents moments del dia.

1.3 Altres aplicacions

El resultat del projecte és realitzar una predicció de disponibilitat de bicicletes en un moment del temps, per un tipus de servei del qual podem emmagatzemar-ne les dades per tal de poder fer-ne una anàlisi. La principal característica del Bicing és que no se'n pot fer una reserva prèvia.

A més, el servei ha de tenir una certa coherència, és a dir, que tingui una demanda continuada. Per exemple, si volem poder predir si quedaran entrades a la venda per un determinat espectacle això dependrà de l'obra que facin, és a dir, que la predicció sobre dades antigues segurament no ens serviria.

Així doncs, l'aplicació d'aquesta arquitectura és per:

- qualsevol servei repetitiu
- del que hi hagi un nombre finit de recursos
- del que no es facin reserves prèvies
- del que es disposin de dades accessibles pel públic

Aplicacions possibles:

- La idea del projecte es pot aplicar a altres serveis de bicicletes compartides a altres ciutats. Hi ha més de 50 ciutats al món amb serveis de bicicletes compartides. Les característiques dels serveis són similars, es basen en estacions, i en disponibilitats de bicicletes i llocs disponibles.
- A Catalunya hi trobem altres serveis, com la Girocleta de Girona o l'Ambicia't de Granollers
- A Espanya hi ha més de 30 ciutats amb serveis similars (entre d'altres: Madrid, Alacant, València, Bilbao, ...)
- A Europa en trobem a 20 països (entre d'altres: Anglaterra, Rússia, Alemanya, ...)
- Serveis de vehicles compartits en general, com motos i cotxes.

Per altra banda, amb poques modificacions en la manera de predir, encara que amb alguns importants en la interfície, podríem fer servir la tecnologia de predicció per tal de donar cobertura a serveis que el pool sigui únic, és a dir d'una única estació. Així doncs podríem predir:

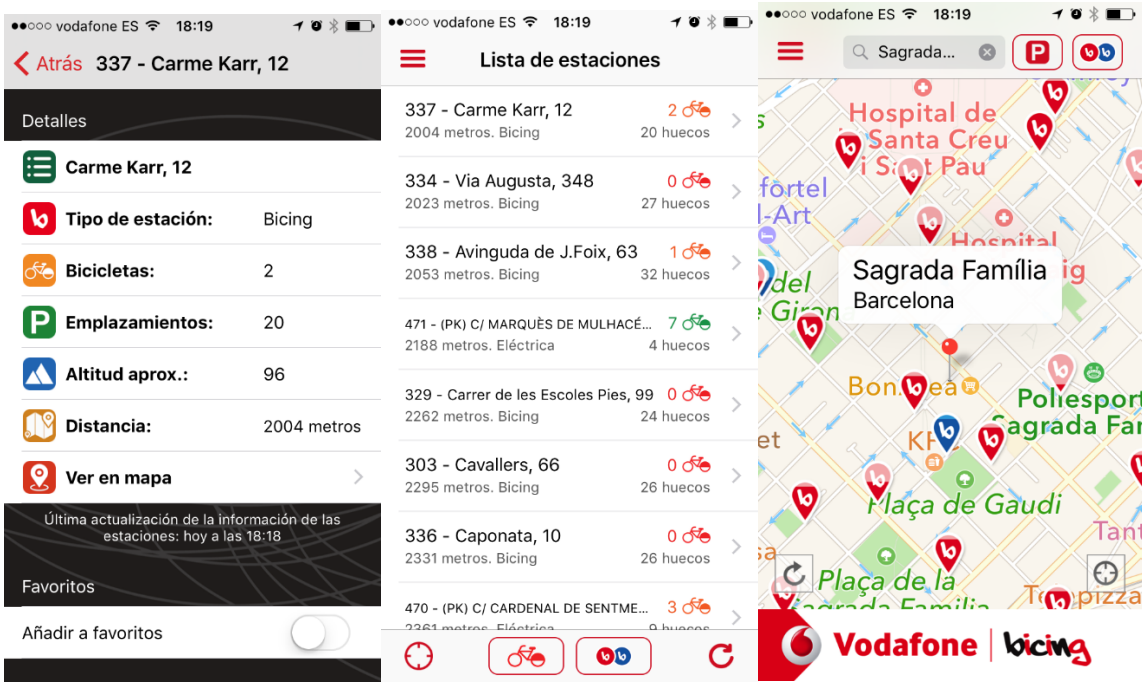
- Restaurants on no fan reserva prèvia de taula.
- Transbordadors.
- Serveis de pàrquing, on ens podem informar de disponibilitats previstes de llocs per aparcar en una determinada zona.

1.4 Estudi de Mercat

Existeixen diverses aplicacions que proporcionen informació del servei de Bicing.

Bicing (Ajuntament de Barcelona)

És l'aplicació oficial del servei de Bicing i una de les més populars. No té utilitat de predicció. Dóna la disponibilitat actual de les estacions. A més, recentment s'hi ha introduït un sistema de *gamificació* per incentivar-ne l'ús.



Citymapper (Citymapper Limited)

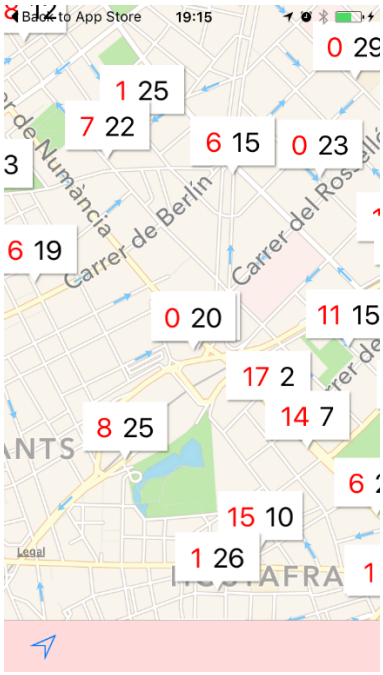
És una aplicació genèrica de transports, no només del Bicing, i cobreix diverses ciutats del món. És una guia per anar d'un punt a l'altre de la ciutat, i permet escollir el mitjà de transport, sent un d'ells és el Bicing. No dóna predicció de bicicletes. No permet transport multimodal.

RideU (Cecilio Tamarit)

És una aplicació de cerca de bicicletes i llocs d'aparcament. Dóna la possibilitat de localitzar una estació apuntant-hi el dispositiu utilitzant realitat augmentada. No ofereix predicció. Dóna un mapa amb la ubicació de les estacions.

Easy Bicing (Juan Villaescusa)

És una aplicació senzilla que mostra un mapa de la ciutat, amb les estacions, i la disponibilitat de bicicletes i places lliures. No té capacitat de predicció.



City Bikes

És una aplicació que mostra bicicletes i places lliures al mapa, amb la possibilitat de localitzar una estació apuntant el dispositiu amb realitat augmentada.

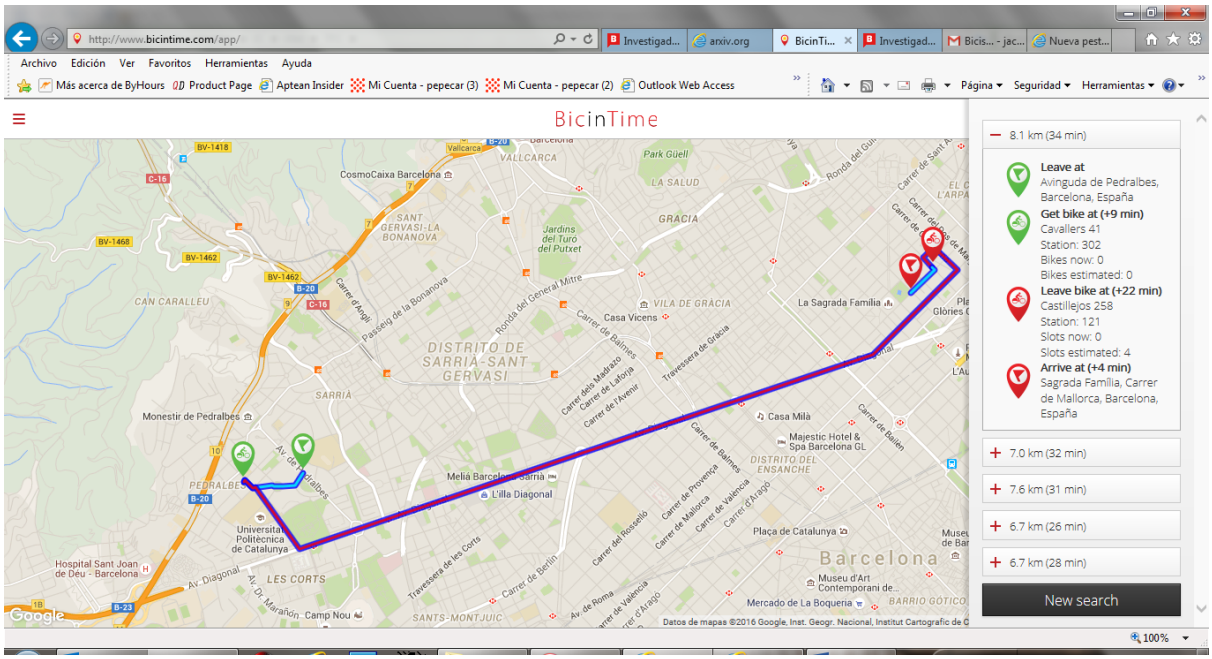
Bicicletes (Jordi Vila)

És una aplicació que mostra bicicletes i places lliures al mapa, amb la possibilitat de localitzar una estació apuntant el dispositiu amb realitat augmentada.

Bicintime

És una aplicació en HTML5, desenvolupada per Gabriel i Alex, estudiants de la UPF. És l'única aplicació que fa predicció. Han publicat informació de l'algorisme de predicció, que es basa en Random Forest. Va estar parcialment subvencionada per la Unió Europea.

A més d'informar del nombre de bicicletes, permet especificar origen i destí, calculant les estacions de recollida i d'entrega de la bici, informant del nombre de bicicletes i de places lliures respectivament.



Quadre Resum

Aplicació	Propietari	Web oficial	Predicció	iOS	Android
Bicing	Ajuntament de Barcelona	https://www.bicing.cat/	No	Sí	Sí
Citymapper	Citymapper Limited	https://citymapper.com/barcelona	No	Sí	Sí
RideU	Cecilio Tamarit	http://cecetaca.com/rideu/en/	No	Sí	Sí
Easy Bicing	Juan Villaescusa	http://easy-bicing.appstor.io/es/	No	Sí	No
City Bikes	Rocket Lab Limited	http://www.citybikesapp.com/	No	Sí	No
Bicicletes	Jordi Vila	http://jordivila.cat/index.php/main/view/bicing/es	No	Sí	No
Bicintime	Gabriel Martins,Alex Bikfalvi (UPF)	http://www.bicintime.com/	Sí	Navegador	Navegador

Competència

L'aplicació més forta i robusta és Bicintime, l'única que fa prediccions sobre la disponibilitat de bicicletes.

De la resta d'aplicacions disponibles no n'hi ha cap que doni una predicció.

Per això, crec que hi ha espai per una altra aplicació que permeti predir la disponibilitat de bicicletes i espais disponibles a una estació. Idealment seria una aplicació nativa per IOS i per Android per poder aprofitar al màxim les característiques dels dispositius.

2 Gestió del projecte.

2.1 Metodologia de desenvolupament

Per desenvolupar el projecte faré servir una metodologia Agile. Les metodologies Agile, una de les més conegudes és SCRUM, es basen en repetides iteracions o esprints. En acabar cadascuna d'elles es fa una avaluació del resultat.

Les metodologies tradicionals de desenvolupament de software es basen en un seguit de tasques que s'executen seqüencialment, un altre nom per anomenar-les metodologies en cascada. Es basa en una definició molt acurada del desenvolupament a realitzar, per arribar a tenir aquesta definició normalment s'estructuren en:

- **Fase d'anàlisi**: on l'usuari expressa els seus requeriments, en definitiva, el que desitja del sistema. Els requeriments han de ser complets i exhaustius, no es basen en una idea o direcció, sinó que és necessari especificar el detall. Típicament estem parlant de que l'usuari:
    - Expliqui la sortida que vol obtenir del sistema, fins i tot indican quina informació vol obtenir en llistats i consultes.
    - Expliqui quines entrades introduirà, o des d'on es poden obtenir.
    - Expliqui quines són les operacions o manipulacions que s'han de fer per tal d'arribar al resultat final.
    - Expliqui les restriccions del sistema quant a permisos, seguretat, verificacions, etc…
    - Fins i tot en certes circumstàncies s'ha d'especificar el joc de proves que s'ha d'executar per tal de verificar el resultat final del sistema.
- **Fase de construcció**: aquí el rol determinant el té l'equip de desenvolupament, que ha d'escriure el disseny del sistema per tal de donar cobertura als requeriments de l'usuari. Aquesta fase típicament es caracteritza per dues tasques:
    - Un disseny en detall de la solució a implementar, incloent-hi els diferents programes o components del sistema, i el disseny del model de dades que es farà servir.
    - El desenvolupament del programari, un cop s'ha verificat amb l'usuari que el disseny plantejat permet assolir els requeriments.

Les metodologies Agile es basen en iteracions, és a dir, en comptes de fer una seqüència de tasques, es fonamenten en anar repetint les iteracions següents fins a arribar al resultat desitjat:

- Una llista de tasques a desenvolupar prioritzades
- Un període d'execució, anomenat esprint, on s'executen aquestes tasques
- Una avaluació dels resultats, i una re-avaluació dels requeriments.

Dins de les metodologies Agile vull destacar SCRUM, que fa èmfasi en la col·laboració entre membres de l'equip, en una comunicació constant, que permet adaptar els esprints als requeriments de negoci, sovint canviants. Ja no ens trobem en un món estàtic, on les necessitats són eixos que no acostumen a canviar en un període llarg de temps, tampoc ens trobem en un entorn tecnològic que dificulta els canvis. SCRUM es basa en una re-definició de les tasques a fer, en permetre canviar el paradigma del software desenvolupat, en tenir resultats des del primer moment, en mantenir el

sistema permanentment disponible, en una re-avaluació tant dels requisits com dels resultats, en una comunicació constant entre els membres. Els rols típics són:

- Product Owner-propietari del projecte, típicament el responsable del departament que ha sol·licitat l'aplicació, i que ha de comunicar a tots els components de l'equip quina és l'estratègia i la visió que l'aplicació ha d'assolir.
- Equip de Projecte: el conjunt de desenvolupadors i dissenyadors que són els que porten a terme les tasques de programació i disseny.
- Scrum Màster: el facilitador de la comunicació entre l'equip de projecte i el patrocinador. No és el cap de projecte típic d'una metodologia tradicional, és el responsable d'incentivar que el projecte avanci.

La documentació que es genera amb la metodologia SCRUM consisteix en:

- Backlog de producte: les especificacions d'alt nivell que el sistema ha d'assolir.
- Backlog d'esprint: la llista prioritzada de tasques a realitzar en el següent esprint.
- Increment de funcionalitat: la funcionalitat del producte assolida després de cada esprint.
- Les històries dels usuaris, són les especificacions que els usuaris demanen, i que són la base del disseny. Cada història d'usuari s'ha de poder escriure en una nota tipus post-it, i hauria de contenir les següents dades:
    - Categoria: Categoria del tipus de història, si es refereix a un requisit d'interfície, funcional, etc
    - Títol: Títol descriptiu de la història
    - Descripció: descripció detallada de la història
    - Prioritat: Priorització de la història respecte a les altres històries del backlog
    - Instruccions per validar: indicacions que confirmaran que el desenvolupament és correcte

Metodologia del projecte

En tractar-se d'un projecte amb un equip de treball format per una sola persona no he tingut ocasió d'aplicar una metodologia SCRUM pura. El que he fet és basar-me en:

- Una llista de temes a desenvolupar, prioritzada
- Desenvolupament del programari
- Avaluació dels resultats.

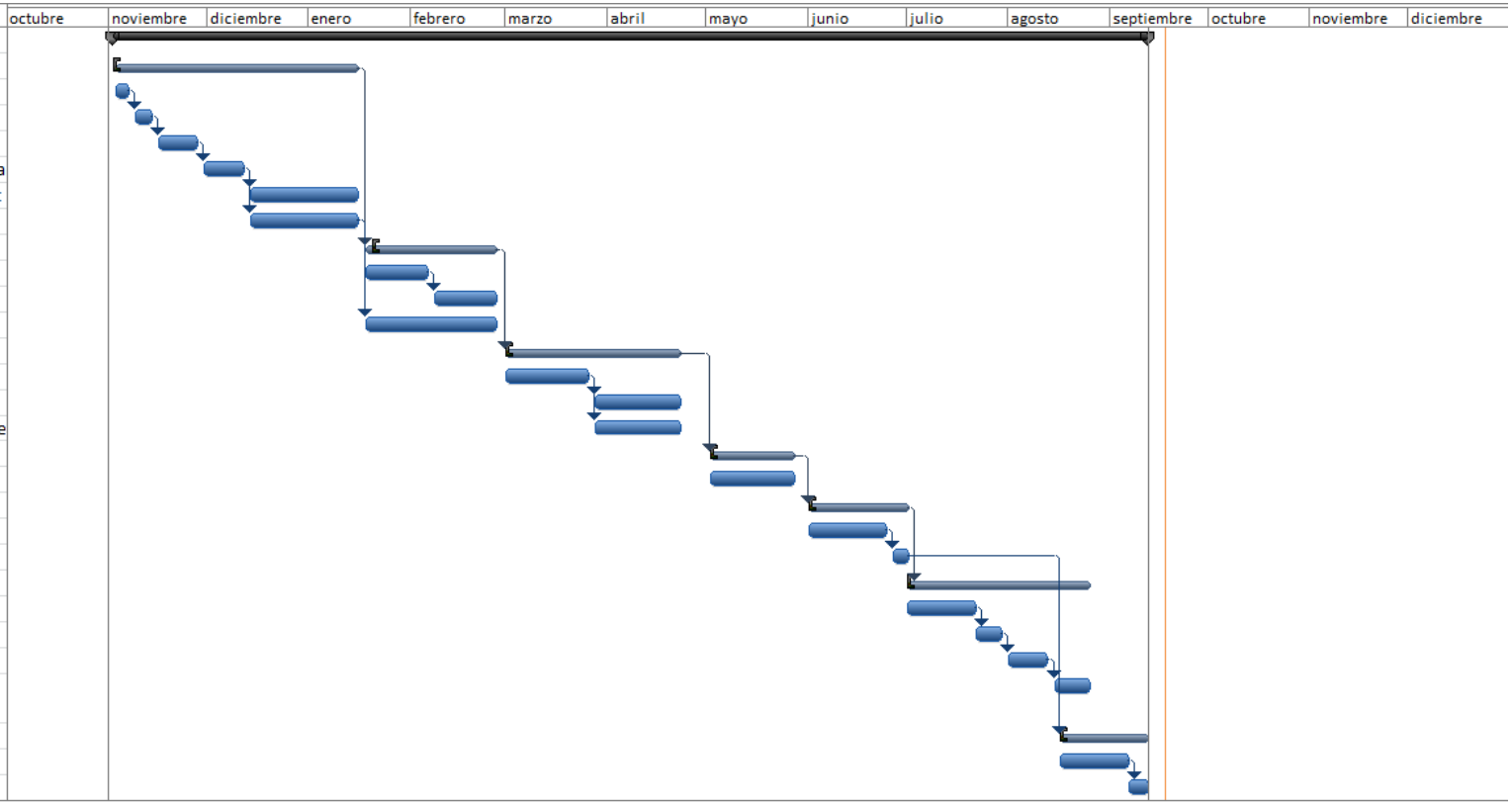
Així mateix he treballat construint un prototip on el sistema:

- S'anés perfeccionant en cada iteració, començant amb funcionalitats bàsiques, i afegint complexitat cada vegada.
- Mantenir el sistema utilitzable des del primer moment.

2.2 Pla del projecte

Esprint	Nom	Data	Hores	Tasca	Hores
1	Definició	02/11/2015	112	Definició àmbit projecte	16
				Requeriments usuaris	16
				Estudi de mercat	16
				Definició components del sistema	16
				Elecció d'eines desenvolupament	24
				Elecció premisses cloud	24
2	Desenvolupament 1	20/01/2016	64	Desenvolupament interfície	32
				Interfície amb mapes	16
				Elaboració informe previ	16
3	Dades	01/03/2016	72	Implementació base de dades	24
				Captura dades Bicing	24
				Captura de dades meteorològiques	24
4	Documentació 1	02/05/2016	24	Elaboració esborrany treball	24
5	Predicció 1	01/06/2016	80	Prediccions regressió lineal	40
				Prediccions regressió polinòmica	40
6	Predicció 2	01/07/2016	132	Implementació Random Forest	60
				Creació containers	16
				Implementació sistema	16
				Proves comparatives precisió predicció	40
7	Documentació 2	16/08/2016	56	Finalització treball	40
				Elaboració presentació	16
				Total Estimat	**540**

Id	Nombre de Tarea
1	**Bicing Stats**
2	Esprint 1 - Definició
3	Definició àmbit projecte
4	Requeriments usuaris
5	Estudi de mercat
6	Definició components del sistema
7	Elecció d'eines desenvolupament
8	Elecció premises cloud
9	Esprint 2 - Desenvolupament 1
10	Desenvolupament interfície
11	Interfície amb mapes
12	Elaboració informe previ
13	Esprint 3 - Dades
14	Implementació base de dades
15	Captura dades bicing
16	Captura de dades meteorològique
17	Esprint 4 - Documentació 1
18	Elaboració esborrany treball
19	Esprint 5 - Predicció 1
20	Prediccions regressió lineal
21	Prediccions regressió polinòmica
22	Esprint 6 - Predicció 2
23	Implementació random forest
24	Creació containers
25	Implementació sistema
26	Proves comparatives precisió predicció
27	Esprint 7 - Documentació 2
28	Finalització treball
29	Elaboració presentació



3 Requisits

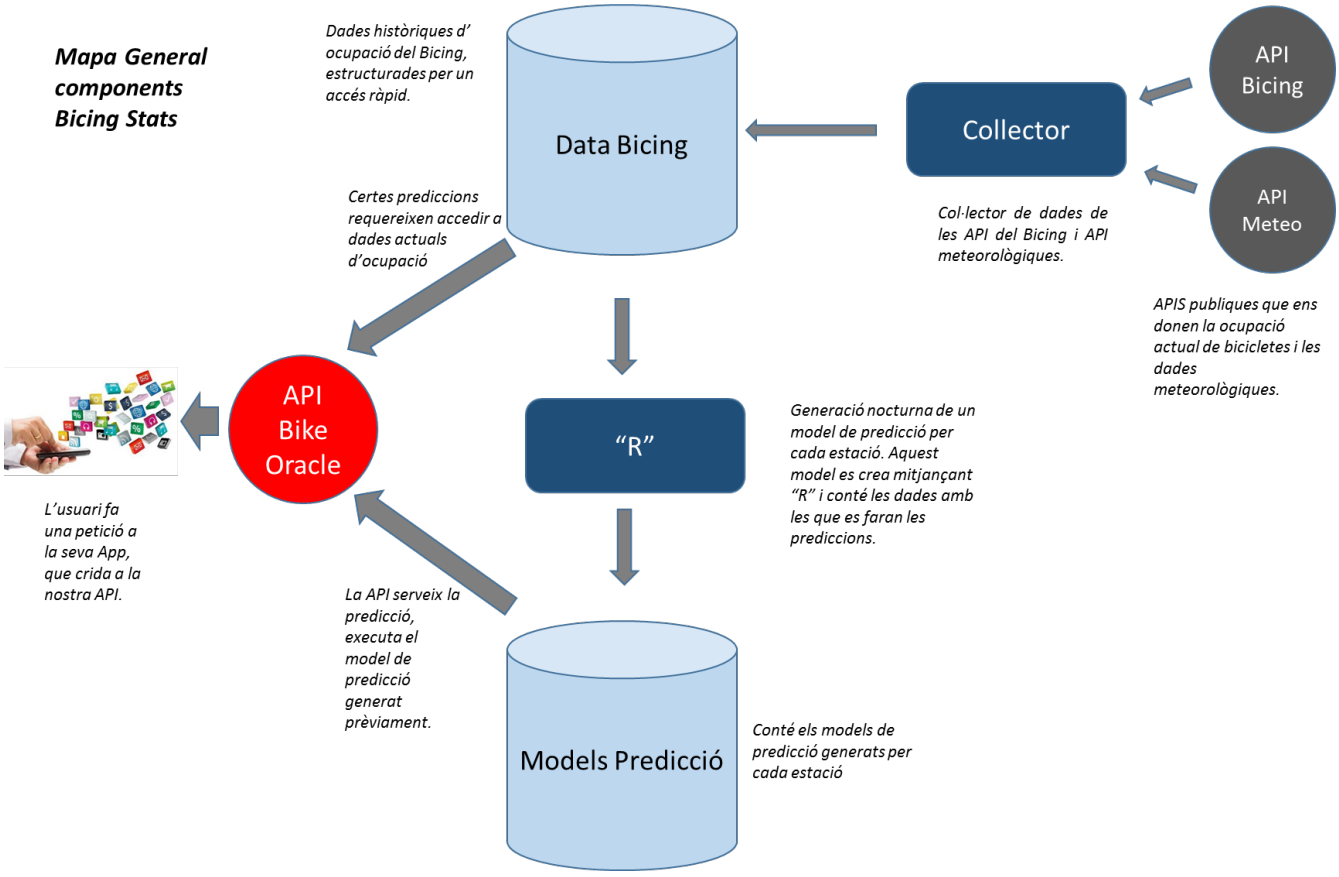
Categoria	Títol	Descripció	Prioritat	Validació
Interfície	Smartphone	Com a usuari, vull poder accedir a l'aplicació des de qualsevol Smartphone	3	He de poder executar l'aplicació en entorns iOS i Android.
Interfície	Nativa	Com a usuari, vull que l'aplicació sigui nativa del dispositiu, és a dir, que pugui aprofitar les característiques del dispositiu on s'executi	1	He de poder instal·lar l'aplicació en el dispositiu des de les plataformes de software de iOS i Android
Interfície	Navegador	Com a usuari, vull poder executar l'aplicació des del meu ordinador personal, en qualsevol navegador.	1	He de poder accedir a l'aplicació des dels principals navegadors.
Interfície	Històric de consultes	Com a usuari, vull guardar les ultimes consultes fetes d'estacions de Bicing	1	Que quan vagi a posar una estació, pugui veure les anteriors estacions consultades prèviament i seleccionar-les.
Interfície	Mapa selecció	Com a usuari, que pugui seleccionar l'estació en un mapa	2	Que pugui marcar un punt en el mapa, i això sigui el meu punt de partida per a buscar estacions

Categoria	Títol	Descripció	Prioritat	Validació
Interfície	Zona	Com a usuari, que el sistema em torni els resultats de bicicletes en les estacions més properes al punt marcat.	2	Que em surtin les estacions properes al punt marcat
Interfície	Horari	Com a usuari, que pugui escollir en quin moment i en quin dia vull saber la probabilitat de trobar bicicleta	3	Que em doni la probabilitat de trobar bicicleta en qualsevol moment que jo esculli
Càlcul	Estació de destí	Com a usuari, voldria poder informar de l'estació de destí on vull anar	3	Que el sistema em demani una estació de destí.
Càlcul	Temps de Viatge	Que l'app em digui el que trigaré pedalant d'una a altra estació	2	Que el sistema em doni el temps de trajecte estimat a la velocitat de bicicleta
Càlcul	Predicció encertada	Com usuari, voldria que la predicció fos acurada, tenint en compte la història de l'ocupació de bicicletes de l'estació	3	Que hi hagi un percentatge d'encerts superior al 70%
Càlcul	Predicció a curt termini	Com a usuari, voldria que l'app m'informés de l'ocupació actual de bicicletes de l'estació	3	Que em doni el nombre de bicicletes que ara mateix hi ha a l'estació

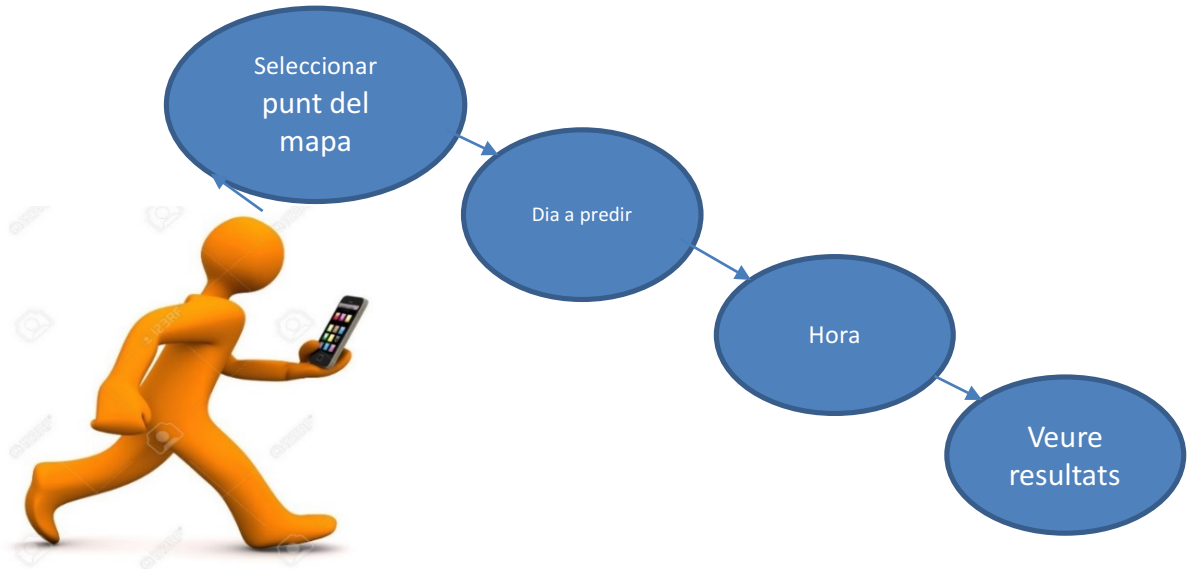
Categoria	Títol	Descripció	Prioritat	Validació
Càlcul	Predicció esdeveniments	L'app haurà de tenir en compte si hi ha algun esdeveniment especial que pugui alterar les bicicletes disponibles	3	Que hi hagi un percentatge d'encerts superior al 70%
Tècnics	Rapidesa	Com a usuari, vull una predicció immediata	3	Que el temps del resultat sigui de menys de 3 segons
Tècnics	Disponibilitat	Com a usuari, vull que l'aplicació estigui sempre disponible	3	Que 95 de cada 100 cops que executo l'aplicació, estigui funcionant.
Funcional	Feedback usabilitat	Com a owner, vull saber si l'usuari està content amb l'arquitectura de l'app	3	Que el sistema pregunti a l'usuari de manera aleatòria cada N usos.

4 Arquitectura

4.1 Components del Sistema

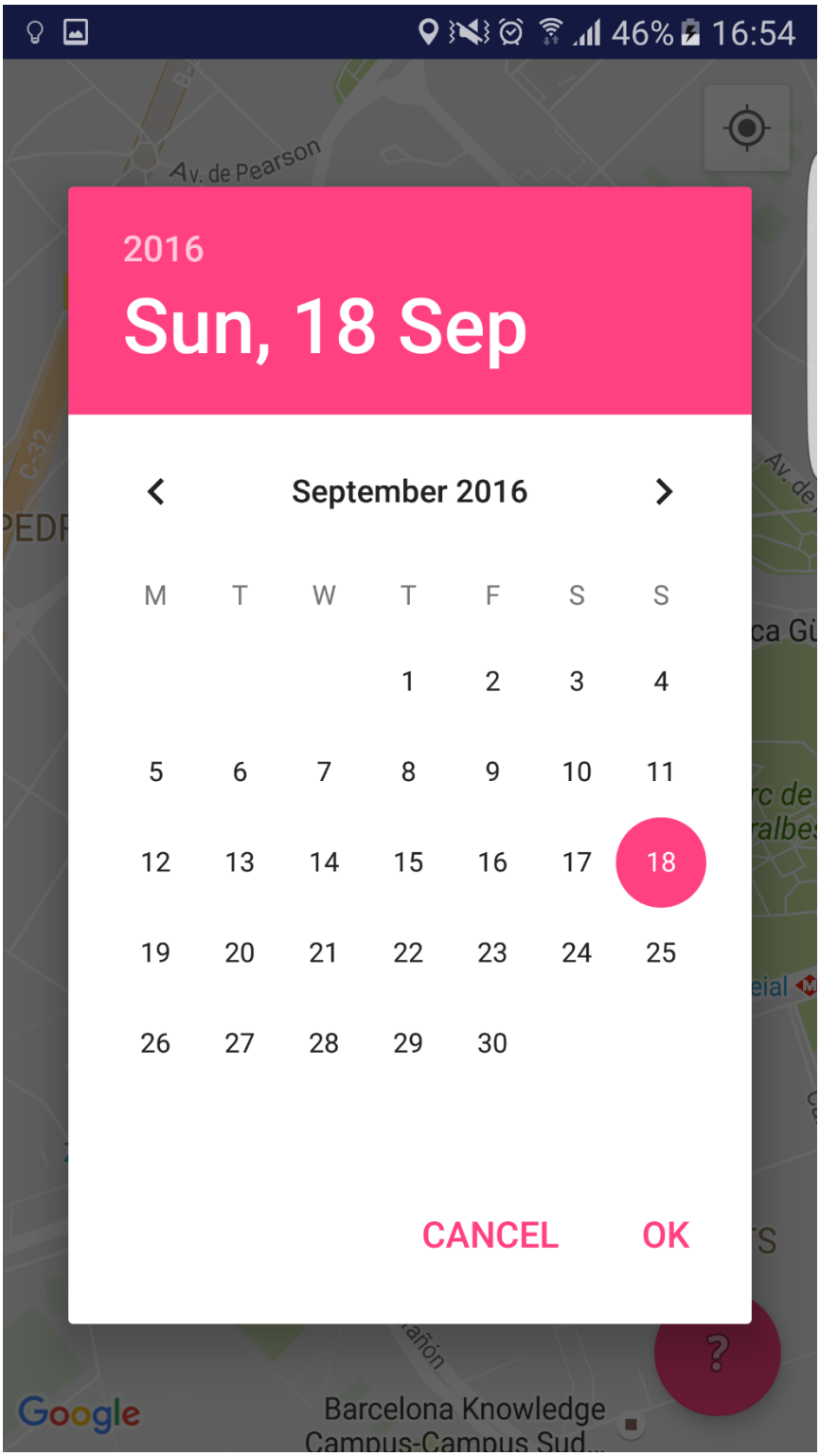


4.2 Casos d'ús

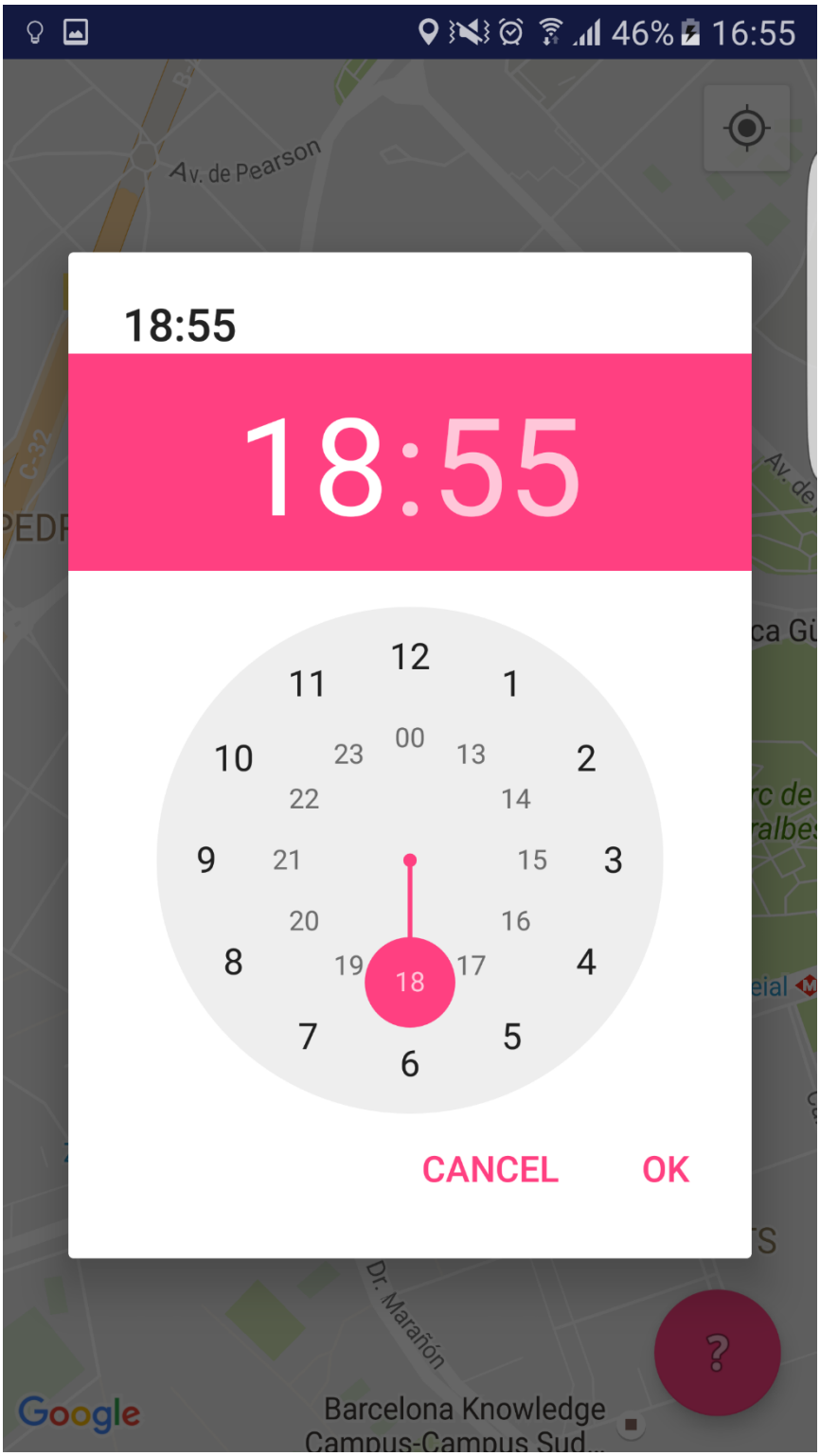


Seleccionar Punt del Mapa

Introduir dia a predir



Introduir la hora



Mostrem els resultats

4.3 Arquitectura de la informació

En el món de les aplicacions mòbils, el disseny de la interfície és decisiu en l'èxit de l'aplicació. La interfície d'usuari és el component més decisiu, encara que no l'únic, que ens determinarà l'experiència de l'usuari.

- L'experiència d'usuari (UX) és el concepte clau de l'aplicació. Busquem:
    - Senzillesa, concepte KISS, sense escenografies exagerades, que tingui en compte el concepte utilitari de la nostra aplicació.
    - Facilitat d'utilització, creiem que l'usuari pot estar caminant pel carrer quan vol comprovar si en una estació hi ha disponibilitat.
    - Que en cada moment l'usuari tingui una visió de les opcions disponibles, i cap on ha d'anar.
- El disseny de la interfície serà inicialment molt simple, més endavant es podran afegir elements visuals que, sense ser una còpia, donin a entendre la relació de l'aplicació amb Bicing. Tractarem d'aprofitar els estàndards de l'arquitectura d'Android.

Per tal de centrar l'abast i les expectatives dels possibles usuaris de l'aplicació, hem tractat de definir els criteris arquitectònics de l'aplicació.

Objecte	**Proveir un servei de predicció de disponibilitat de bicicletes del Bicing.**
Dirigit a	Usuaris del sistema Bicing. • Abonats: 98.497 • Utilitzacions mensuals: 1.027.707 • Mitjana d'usos diaris: 3.333
Interacció	Creiem que els usuaris faran servir l'aplicació des d'un dispositiu mòbil abans de trobar una bicicleta, normalment quan vagin pel carrer. Per això buscarem una interacció el més senzilla possible, amb botons grans, centrats a la pantalla, i amb valors per defecte de les peticions d'informació.
Navegació	Atès els casos d'ús de l'aplicació implementarem una navegació molt senzilla, sense més casuística que passar de plànol-dia-hora i plànol de resultats.
Continguts	En aquesta primera fase de l'aplicació no tindrem cap contingut.
Alertes	En aquesta primera fase de l'aplicació no tindrem cap alerta.
Disseny gràfic	En aquesta primera fase no tenim pensant més que unes pantalles molt senzilles, basant-nos en l'estàndard d'Android. En una fase posterior buscarem un patró de disseny gràfic que recordi el vermell i blanc del Bicing amb alguna característica distintiva que tracti de suggerir predicció i encert.
Feedback	Implementarem un sistema de feedback per tal que l'usuari pugui informar de possibles problemes d'usabilitat

4.4 Base de dades

La plataforma on el sistema emmagatzema les diferents dades que requereix per tal de procedir a fer els càlculs de probabilitat.

Una de les tasques que he hagut de fer és escollir un motor de base de dades i en quina modalitat vull fer-ho, si en una màquina a les meves instal·lacions, o bé en un servei de hosting.

Requeriments

- Voldrem poder accedir a un joc de dades concret: facilitat per tal d'indexar les taules i poder accedir de manera directa.
- Voldrem utilitzar llenguatge SQL, per la seva facilitat d'utilització, la seva popularitat, i la disponibilitat i facilitat d'utilització des de les eines de programació escollides.
- Voldrem poder accedir a les dades meteorològiques relacionades amb una estació concreta.
- No requerirem una especial potencia de funcionalitat de modificació de dades, sent les operacions més freqüents les d'inserció i consulta.

Dades a emmagatzemar

Les dades que emmagatzemem seran les que farem servir per tal d'estimar la predicció, i són:

- Dades d'ocupació del Bicing: les dades d'ocupació real del Bicing, per cada estació.
- Dades meteorològiques: la quantitat de pluja categoritzada i la temperatura,
- Dades de les peticions rebudes: emmagatzemem les peticions que ens estan fent els usuaris, així com els resultats obtinguts.
- Dades de les peticions d'autoaprenentatge que el sistema provoca, les característiques de la mostra, i el resultat.

Dimensió.

En aquest punt del projecte ens interessa fer una estimació del volum de dades que tindrem, encara no tenim el model de dades definit, així que farem una estimació inicial. Aquesta estimació ens ha de servir per escollir el gestor de base de dades.

El dataset més gran serà el de dades d'ocupació de Bicing. L'API que Bicing ofereix, té el següent model de dades:

Dada	Bytes
Id	4
status	1
slots	2
bikes	2
timestamp	4
Total	13

És a dir, que per cada lectura que fem al dia, de cada estació, necessitarem 13 bytes. Així doncs segons:

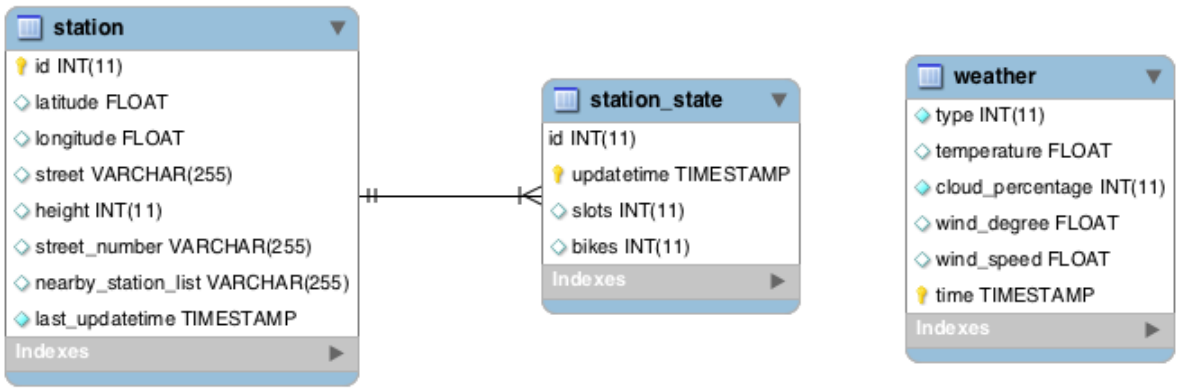
Estacions	500
Lectures diàries	1440

Hem estimat:

Dades/dia (MB)	9,14
Dades/any (MB)	3336,33

A més d'aquestes dades, requerirem espai addicional pels altres datasets, així que calculem que estarem al voltant de 4000 MB/any.

Model de Dades



4.5 Alternatives d'implementació

On Premise envers Cloud

La primera decisió ha estat avaluar si instal·laria la base de dades en un ordinador propi a les meves instal·lacions (On Premise), o bé un servei extern (Cloud).

- **On Premise**: La base de dades, motor i dades, estan ubicades a un servidor propi, a les instal·lacions d'una oficina de BicingStats (de moment a casa meva), les tasques d'explotació del sistema las portaria a terme personal propi de Bicing Stats (de moment jo mateix). El sistema no és escalable, si el programa es popularitza i el fa servir molta gent hauria d'instal·lar un servidor addicional per tal que la resposta continuï sent ràpida. A mesura que vaig afegint història a la base de dades, és possible que requereixi espai d'emmagatzematge addicional, és a dir, que hauria d'afegir unitats de disc.
- **Cloud**: La base de dades està ubicada a un servidor extern, el servidor està en un servei cloud, on tenen personal propi per fer les tasques de manteniment. Allà instal·laria el programari necessari igual que podria fer-ho a la meva màquina personal. Podria disposar de CPUs, memòria o emmagatzematge addicional en cas que ho requerís.

Taula comparativa

Servei	On Premise	Cloud
Espai físic	Espai propi de l'empresa.	Data Center
Comunicacions	Gestionades per l'empresa.	Gestionades pel data center. Normalment ampli de banda escalable.
Servidor	Propi	Propietat del data center
Gestió del servidor (actualitzacions, backup)	Personal propi	Totalment automatitzat.
Escalabilitat	Responsabilitat dels tècnics de l'empresa que han de preveure puntes d'utilització i dimensionar els recursos segons aquestes demandes.	Depenent del data center serà més o menys automàtic

Alternatives considerades

Servei	Característiques	Preu	Pros	Contra
Digital Ocean	MySQL autogestionat amb la creació d'un entorn amb docker.	6 $/mes (20 GB + backups) 12 $/mes (30 GB + backups) -> gratis per tenir compte d'estudiant fins a 100 $	Ja tinc experiència prèvia amb mysql, lo que facilitarà la implementació. És Open Source Utilitzat per tot tipus d'aplicacions, és ben coneguda la seva robustesa.	Requereix monitorització
Orchestrate	Solució propietària de base de dades.	50€/mes, gratis essent estudiant.	No requereix monitorització.	Solució nova, propietària. No hi tinc experiència prèvia.
Redshift	Gestionat per Amazon	Tarifa bàsica 216€/mes	Monitoritzat per Amazon API PostgreSql Especialitzat en Big Data	Tarifa molt alta
Relational Database Service (RDS)	Gestionat per Amazon. MySql (encara que té altres solucions).	A partir de 12€		

Decisió Final

He escollit l'opció de Digital Ocean:

- Funcionament molt simple, permet crear instàncies en les que tenim accés de root, pel que hi podem instal·lar qualsevol utilitat que ens faci falta
- Tarifa molt ajustada: permet modificar la capacitat de les instàncies fàcilment i en poc temps (~10 minuts), amb una tarifa inicial de 5 $ al mes.

Característiques

Nom de la instància: docker-alex

Servidor: Ubuntu 14.04

Contingut: La infraestructura està dividida en diversos contenidors amb Docker, que permeten aïllar els diversos components del sistema i assegurar que els entorns on s'executa el codi a producció són els mateixos que en local.

4.6 Entorn de desenvolupament

Desplegament:

Faig servir Docker com a sistema de contenidors per assegurar compatibilitat dels entorns de desenvolupament i producció. Docker és una eina de desplegament multi plataforma, que permet distribuir i desenvolupar.

Llenguatges de programació

Servidor

He fet servir Go per recol·lectar dades i servir les peticions. Go és un llenguatge modern, que dóna molta facilitat per crear apis i que és capaç de proporcionar un rendiment força alt i que facilita paral·lelitzar tasques.

> Inicialment vaig començar un primer prototip en PHP (https://github.com/alexcarol/bicing-stats), que conec bé, però vaig topar amb alguns problemes de memòria al implementar la recol·lecció de dades, que encara que probablement hauria pogut solucionar, vaig preferir canviar a un llenguatge més eficient, per evitar encallar-me amb limitacions del llenguatge.

Per implementar el model de predicció he fet servir el llenguatge R, per tal d'analitzar les dades recol·lectades i fer prediccions sobre les peticions. R és un llenguatge utilitzat per l'obtenció de càlculs i gràfics estadístics, pel que permet iterar molt ràpidament i que a més proporciona un rendiment força bo tenint en compte que és un llenguatge interpretat.

> Més endavant podria plantejar substituir R per una implementació en Go dels algorismes que utilitzo o potser un altre llenguatge com Python, que també té moltes llibreries estadístiques.

Client (app Android)

He fet servir Java, que proporciona moltes llibreries per implementar la interfície gràfica. Per altra banda, en un primer moment vaig considerar fer l'aplicació en Go, per tal de poder compartir codi amb el servidor, però vaig desistir-ne perquè tal com he plantejat l'aplicació mai hauré d'implementar la mateixa lògica al client i al servidor (seria diferent si per exemple estigués implementant un joc amb servidor autoritari, per exemple), a més, Go no disposa de moltes de les facilitats que dóna Java a l'hora de pintar la interfície gràfica.

Base de dades

He triat MySQL per emmagatzemar les dades que es fan servir per realitzar les prediccions. L'he triat perquè és una solució que conec, que es fa servir àmpliament i que té un rendiment bo.

Editors

Atom per la part de servidor, és un editor creat per Github i té infinitat de plugins per facilitar el desenvolupament amb la majoria de llenguatges, el que el fa apte per programar pràcticament qualsevol projecte.

Android Studio per la part de client. Aquest editor està basat en el reconegut IntelliJ IDEA, de JetBrains, ja que incorpora tot el necessari per crear i provar una app Android, incloent-hi el desplegament en un dispositiu Android o l'emulació en local.

Sistema de Control de Versions

Com a control de versions he decidit fer servir Git, un sistema de control de versions distribuït creat per Linus Torvalds. El codi es troba allotjat a GitHub, que és un servei d'allotjament de repositoris git que és gratuït per projectes Open Source, com el meu. El codi l'he dividit en dos repositoris: https://github.com/alexcarol/bicing-oracle-app (app Android) i https://github.com/alexcarol/bicing-oracle (servidor).

5 Motor de Predicció

El motor de predicció és el nucli de l'aplicació, ja que ens ha d'aportar el valor afegit al programa, per sobre de la informació de les dades actuals.

Per tal de simplificar les explicacions, considerarem a partir d'ara que el que busquem és una bicicleta en una estació determinada.

Les dades històriques ens poden servir per apuntar tendències de l'estació, però el seu pes haurà de tenir en compte diferents factors de pertorbació:

1. Dia de la setmana

2. Factor meteorològic

5.1 Model De Regressió Lineal

Per tal d'escollir el model estadístic de regressió he estat fent diferents proves.

La primera consisteix en regressió lineal amb dades d'ocupació d'una estació durant un període de dos mesos. Per fer aquesta regressió he emprat el programa R commander.

Prova 1

Per aquesta prova he emprat tres variables:

- Day.moment (temps transcorregut des de les 00:00)
- "Labor", una variable discreta que indica si el dia és laborable.
- Time, que es representa com a Epoch time (segons passats des de l'1 de gener de 1970).

Les tres variables tenen un p-valor molt petit (menor a $2*10^{-16}$), pel que són estadísticament rellevants. El model té un coeficient de determinació (r^2) d'aproximadament un 8%, pel que la fiabilitat d'aquest model aplicat a la predicció no és gaire significant.

```
Call:

lm(formula = bikes ~ Day.moment + Labor + time, data = Datos)

Residuals:

     Min       1Q   Median       3Q      Max

-17.0691  -4.8122   0.5537   5.2220  16.1872
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.226e+03  2.666e+01   45.97   <2e-16 ***
Day.moment  -6.314e-05  9.596e-07  -65.80   <2e-16 ***
Labor       -1.143e+00  5.545e-02  -20.61   <2e-16 ***
time        -8.301e-07  1.830e-08  -45.37   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 6.655 on 75282 degrees of freedom
Multiple R-squared:  0.08326, Adjusted R-squared:  0.08322
F-statistic:  2279 on 3 and 75282 DF,  p-value: < 2.2e-16
```

Prova 2

Al veure que el primer model no era estadísticament significant he provat amb un segon model, en que fem servir time i Day.moment (que ja fèiem servir al model anterior) i en comptes de Labor hem optat per fer servir la variable qualitativa "dia de la setmana", que la dividirem en 6 variables fictícies.

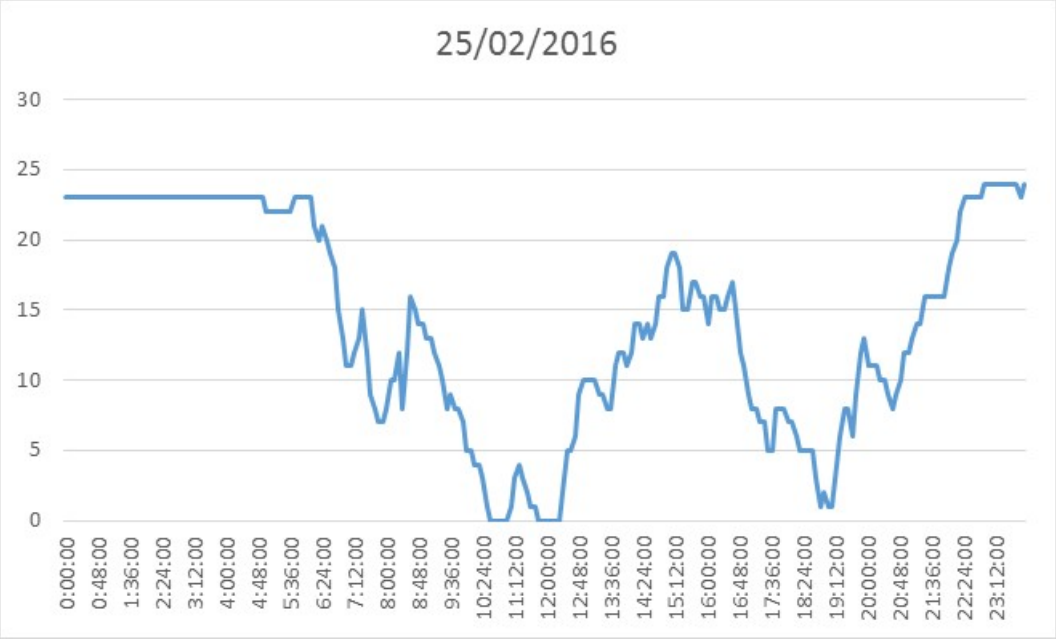
Una vegada més, hem pogut veure que el p-valor indica que les variables són rellevants. El coeficient de determinació és d'aproximadament el 9%, lleugerament millor que en el cas anterior, però segueix sense poder ser aplicat per a prediccions.

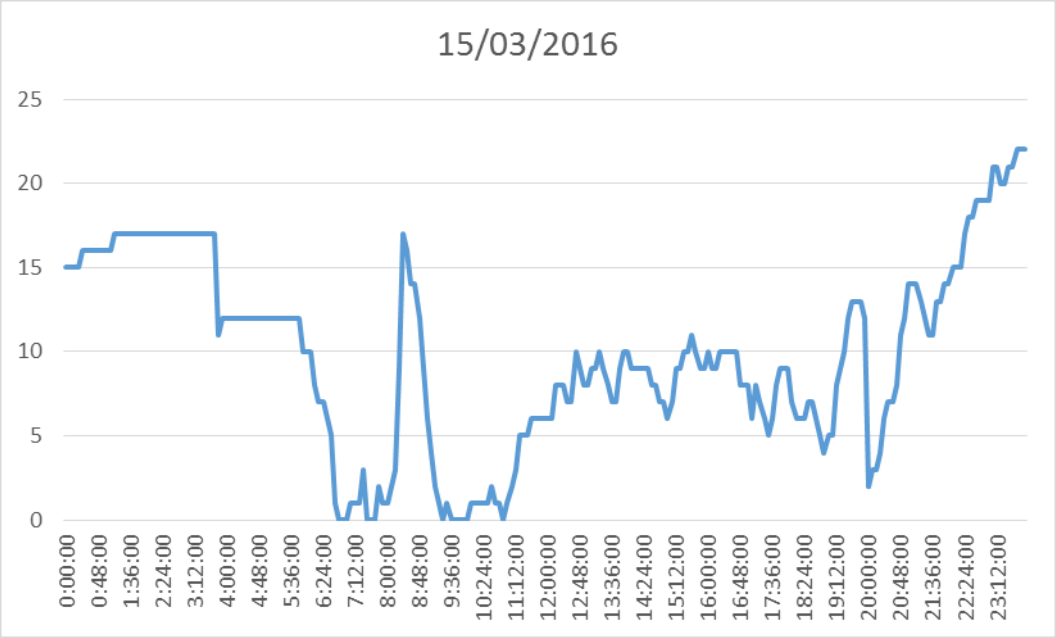
```
Call:
lm(formula = bikes ~ Day.moment + time + Lunes + Martes + Miércoles +
    Jueves + Viernes + Sábado, data = Datos)
Residuals:
     Min       1Q   Median       3Q      Max
-17.5257  -4.8218   0.4113   5.1860  16.3727
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.204e+03  2.663e+01   45.21   <2e-16 ***
Day.moment  -6.343e-05  9.564e-07  -66.32   <2e-16 ***
time        -8.146e-07  1.827e-08  -44.58   <2e-16 ***
Lunes       -9.979e-01  8.957e-02  -11.14   <2e-16 ***
```

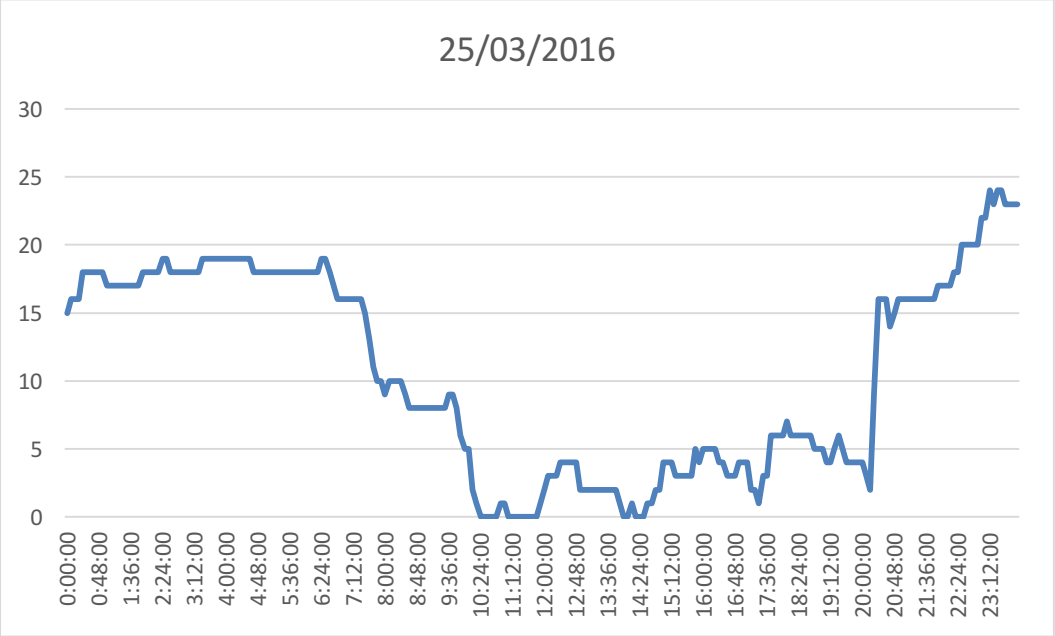
```
Martes       -1.578e+00  8.946e-02  -17.64   <2e-16 ***
Miércoles    -9.843e-01  8.963e-02  -10.98   <2e-16 ***
Jueves       -2.349e+00  8.984e-02  -26.14   <2e-16 ***
Viernes      -2.261e+00  9.253e-02  -24.43   <2e-16 ***
Sábado       -1.033e+00  9.550e-02  -10.82   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 6.631 on 75277 degrees of freedom
Multiple R-squared:  0.08995, Adjusted R-squared:  0.08985
F-statistic: 930.1 on 8 and 75277 DF,  p-value: < 2.2e-16
```

Conclusions

Si fem un gràfic de les dades de l'estació:

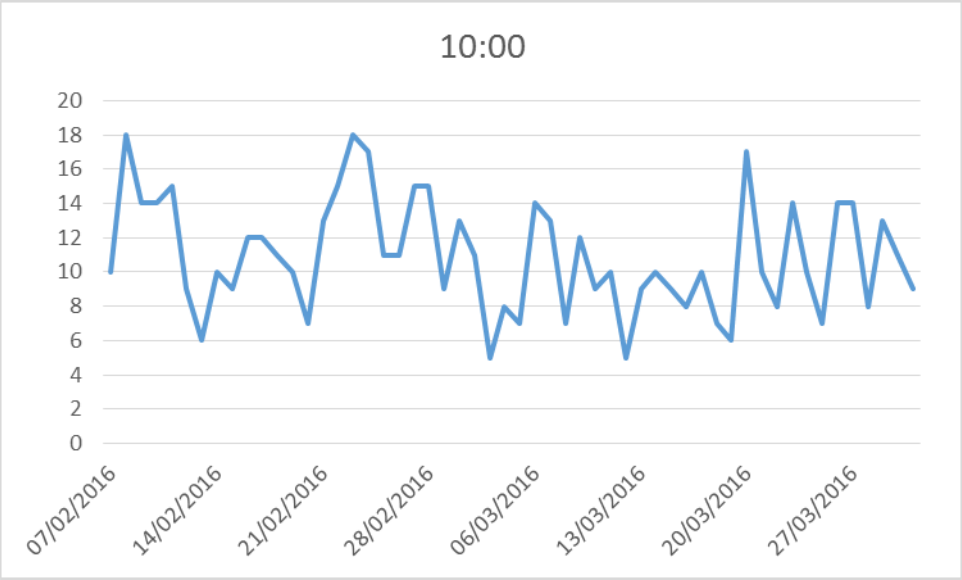


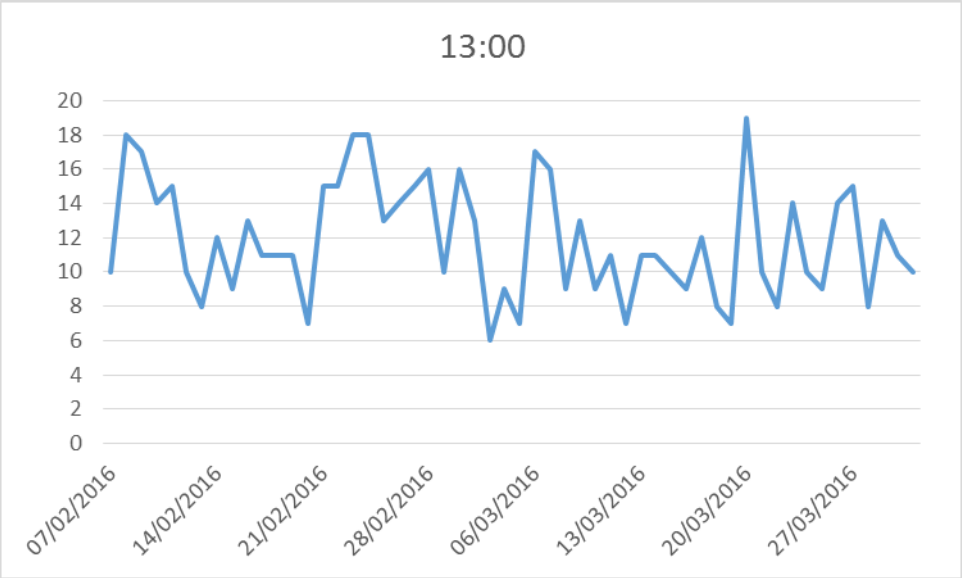




En els gràfics es pot observar que no sembla que les dades de les estacions segueixin un patró lineal.

També he tractat d'aïllar les dades de les estacions en el mateix rang horari. Els resultats són:





El gràfic mostra les dades mitges de la hora, per cada dia del mes. Observem que les dades mitges no són un bon índex de predicció, doncs sempre ens donarà que tenim bicicletes disponibles.

Veient els resultats d'aquestes proves he decidit buscar un altre model de regressió per realitzar les prediccions.

5.2 Model De Regressió Random Forest

Donat que el model de regressió lineal no ens servia hem buscat un altre que fos més precís. Per això hem triat Random Forest, que és un algorisme que fa servir subconjunts de dades i de criteris de classificació per construir arbres de decisió amb variació controlada.

Hi ha altres algorismes de predicció que podríem emprar, com la regressió logística, però hem optat a fer servir Random Forest perquè dóna suport a variables categòriques (com el temps atmosfèric), mentre que regressió logística no.

He provat diverses variacions per saber si hi hauria bicicletes o no, els factors principals que he considerat són:

- La previsió meteorològica
- El moment del temps pel qual es vol realitzar la predicció
- El dia de la setmana

Hem focalitzat les proves en 4 estacions concretes. Hem fet una predicció en un moment del temps en que ja disposàvem dels resultats, de manera que hem pogut comparar amb la realitat del nombre de bicicletes que teníem.

Els resultats de les prediccions es donen separats per cadascuna de les quatre estacions de mostra, segons els següents criteris:

- Si hi havia bicicletes: en quants casos he encertat que si que n'hi havia.
- Si no hi havia bicicletes: en quants casos he encertat que no n'hi havia.
- També he separat els resultats dels dies feiners i dels caps de setmana. Creiem que l'ocupació de bicicletes els dies feiners obeeix a patrons més pautats de comportament que no pas el cap de setmana.

Vull remarcar que les dades disponibles del Bicing són des del juny, i que no hi ha possibilitat de sol·licitar dades antigues. Aquest fet condiciona l'aprenentatge del model, i a més ens pot induir a conclusions respecte a factors externs que assenyalaré en cada resultat.

Quadre resum principals proves (v)

Variable	Tipus	Descripció	v1	v2	v3	v4bis	v5	v6
pbikes	dependent	indica la probabilitat de trobar o no bicicletes en una estació	X	X	X	X	X	
num_bikes	dependent	nombre de bicicletes a una estació						X
weather_type	categòrica	el tipus de temps que fa (sol, ennuvolat, pluja suau, intensa, ...)	X	X			X	
weather_type	categòrica	variable categòrica booleana que representa si hi ha precipitacions o no			X	X		X
temperature	contínua	la temperatura en graus Celsius				X	X	X
updatetime	contínua	el moment del temps pel qual es vol la predicció	X	X	X	X	X	X
dayMoment	contínua	el temps en segons des de l'inici del dia	X	X	X	X	X	X
weekday	categòrica	el dia de la setmana	X		X	X	X	X
weekday-bis	categòrica	dia de la setmana, dividit en 6 variables "dummy"		X				

v1-Prova bàsica

Variable dependent:

- pbikes: indica la probabilitat de trobar o no bicicletes en una estació

Variables explicatives:

- weather_type: variable categòrica que representa el tipus de temps que fa (sol, ennuvolat, pluja suau, intensa, ...)
- updatetime: variable contínua que representa el moment del temps pel qual es vol la predicció
- dayMoment: variable contínua que representa el temps en segons des de l'inici del dia
- weekday: variable categòrica que representa el dia de la setmana

Resultats

		LABORABLES			FESTIUS			TOTAL		
Est.	**pBikes**	**Encert**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**
30	1	4.725	0	100%	1.369	0	100%	6.094	0	100,0%
30	0	0	75	0%	0	26	0%	0	101	0,0%
42	1	3.828	278	93%	1.258	24	98%	5.086	302	94,4%
42	0	226	468	33%	19	94	17%	245	562	30,4%
74	1	3.943	315	93%	1.268	0	100%	5.211	315	94,3%
74	0	0	574	0%	0	127	0%	0	701	0,0%
366	1	3.855	0	100%	1.387	8	99%	5.242	8	99,8%
366	0	26	919	3%	0	0	#DIV/0!	26	919	2,8%
	1	**16.351**	**593**	**97%**	**5.282**	**32**	**99%**	**21.633**	**625**	**97,2%**
	0	**252**	**2.036**	**11%**	**19**	**247**	**7%**	**271**	**2.283**	**10,6%**
		16.603	**2.629**	**86%**	**5.301**	**279**	**95%**	**21.904**	**2.908**	**88,28%**

Comentaris

Encert molt alt en total, però molt baix en cas que no hi ha bicicleta.

V2-dies de la setmana "dummies"

Variable dependent:

- pbikes: indica la probabilitat de trobar o no bicicletes en una estació

Variables explicatives:

- weather_type: variable categòrica que representa el tipus de temps que fa (sol, ennuvolat, pluja suau, intensa, ...)
- updatetime: variable contínua que representa el moment del temps pel qual es vol la predicció
- dayMoment: variable contínua que representa el temps en segons des de l'inici del dia
- weekday: variable categòrica que representa el dia de la setmana***, en aquest cas l'hem dividit en 6 variables "dummy" per assegurar-nos que l'algorisme de predicció la reconeixia com a variable categòrica.***

Resultats

		LABORABLES			FESTIUS			TOTAL		
Est.	**pBikes**	**Encert**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**
30	1	4.756	0	100%	1.369	0	100%	6.125	0	100,0%
30	0	0	76	0%	0	26	0%	0	102	0,0%
42	1	3.943	195	95%	1.281	1	100%	5.224	196	96,4%
42	0	144	550	21%	0	113	0%	144	663	17,8%
74	1	3.795	463	89%	1.268	0	100%	5.063	463	91,6%
74	0	0	574	0%	0	127	0%	0	701	0,0%
366	1	3.838	49	99%	1.307	88	94%	5.145	137	97,4%
366	0	10	935	1%	0	0	#DIV/0!	10	935	1,1%
	1	**16.332**	**707**	**96%**	**5.225**	**89**	**98%**	**21.557**	**796**	**96,4%**
	0	**154**	**2.135**	**7%**	**0**	**266**	**0%**	**154**	**2.401**	**6,0%**
		16.486	**2.842**	**85%**	**5.225**	**355**	**94%**	**21.711**	**3.197**	**87,2%**

Comentaris

- Encert molt alt en total, però inferior al primer cas. Molt baix en cas que no hi ha bicicleta.

v3-Temps booleà (plou/no plou).

Variable dependent:

- pbikes: indica la probabilitat de trobar o no bicicletes en una estació

Variables explicatives:

- weather_type: ***variable categòrica booleana que representa si hi ha precipitacions o no. A diferència dels casos anteriors aquí hem volgut simplificar aquesta variable per veure si podíem "ajudar" a l'algorisme de predicció***
- updatetime: variable contínua que representa el moment del temps pel qual es vol la predicció
- dayMoment: variable contínua que representa el temps en segons des de l'inici del dia
- weekday: variable categòrica que representa el dia de la setmana

Resultats

		LABORABLES			FESTIUS			TOTAL		
Est.	**pBikes**	**Encert**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**
30	1	4.756	0	100%	1.369	0	100%	6.125	0	100,0%
30	0	0	76	0%	0	26	0%	0	102	0,0%
42	1	3.916	222	95%	1.174	108	92%	5.090	330	93,9%
42	0	190	504	27%	55	58	49%	245	562	30,4%
74	1	4.011	247	94%	1.268	0	100%	5.279	247	95,5%
74	0	0	574	0%	0	127	0%	0	701	0,0%
366	1	3.877	10	100%	1.363	32	98%	5.240	42	99,2%
366	0	20	925	2%	0	0	#DIV/0!	20	925	2,1%
	1	**16.560**	**479**	**97%**	**5.174**	**140**	**97%**	**21.734**	**619**	**97,2%**
	0	**210**	**2.079**	**9%**	**55**	**211**	**21%**	**265**	**2.290**	**10,4%**
		16.770	**2.558**	**87%**	**5.229**	**351**	**94%**	**21.999**	**2.909**	**88,32%**

Comentaris

- Encert molt alt en total, continua sent baix en cas de no haver-hi bicicletes.

v4bis- Temps booleà i temperatura

Variable dependent:

- pbikes: indica la probabilitat de trobar o no bicicletes en una estació

Variables explicatives:

- weather_type: variable categòrica booleana que representa si hi ha precipitacions o no
- ***temperature: variable contínua que representa la temperatura en graus Celsius***
- updatetime: variable contínua que representa el moment del temps pel qual es vol la predicció
- dayMoment: variable contínua que representa el temps en segons des de l'inici del dia
- weekday: variable categòrica que representa el dia de la setmana

Resultats

		LABORABLES			FESTIUS			TOTAL		
Est.	**pBikes**	**Encert**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**
30	1	4.756	0	100%	1.369	0	100%	6.125	0	100,0%
30	0	0	76	0%	0	26	0%	0	102	0,0%
42	1	3.882	256	94%	1.282	0	100%	5.164	256	95,3%
42	0	205	489	30%	0	113	0%	205	602	25,4%
74	1	3.939	319	93%	1.268	0	100%	5.207	319	94,2%
74	0	0	574	0%	0	127	0%	0	701	0,0%
366	1	3.815	72	98%	1.321	74	95%	5.136	146	97,2%
366	0	47	898	5%	0	0	#DIV/0!	47	898	5,0%
	1	**16.392**	**647**	**96%**	**5.240**	**74**	**99%**	**21.632**	**721**	**96,8%**
	0	**252**	**2.037**	**11%**	**0**	**266**	**0%**	**252**	**2.303**	**9,9%**
		16.644	**2.684**	**86%**	**5.240**	**340**	**94%**	**21.884**	**3.024**	**87,9%**

Comentaris

- La introducció de la temperatura no suposa cap diferència respecte als altres models. Aquest factor probablement és més rellevant a l'hivern, que no pas a l'estiu.

v5-Temps categoritzat i Temperatura

Variable dependent:

- pbikes: indica la probabilitat de trobar o no bicicletes en una estació

Variables explicatives:

- ***weather_type: variable categòrica que representa el tipus de temps que fa (sol, ennuvolat, pluja suau, intensa, ...)***
- ***temperature: variable contínua que representa la temperatura en graus Celsius***
- updatetime: variable contínua que representa el moment del temps pel qual es vol la predicció
- dayMoment: variable contínua que representa el temps en segons des de l'inici del dia
- weekday: variable categòrica que representa el dia de la setmana

Resultats

		LABORABLES			FESTIUS			TOTAL		
Est.	**pBikes**	**Encert**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**	**Encerts**	**Errors**	**%Enc.**
30	1	4.756	0	100%	1.369	0	100%	6.125	0	100,0%
30	0	0	76	0%	0	26	0%	0	102	0,0%
42	1	3.863	275	93%	1.282	0	100%	5.145	275	94,9%
42	0	178	516	26%	0	113	0%	178	629	22,1%
74	1	3.964	294	93%	1.268	0	100%	5.232	294	94,7%
74	0	0	574	0%	0	127	0%	0	701	0,0%
366	1	3.813	74	98%	1.324	71	95%	5.137	145	97,3%
366	0	62	883	7%	0	0	#DIV/0!	62	883	6,6%
	1	**16.396**	**643**	**96%**	**5.243**	**71**	**99%**	**21.639**	**714**	**96,8%**
	0	**240**	**2.049**	**10%**	**0**	**266**	**0%**	**240**	**2.315**	**9,4%**
		16.636	**2.692**	**86%**	**5.243**	**337**	**94%**	**21.879**	**3.029**	**87,84%**

- **Comentaris**
  - Resultats molt similar, la temperatura continua sense resultar rellevant.

v6-Predim el nombre de bicicletes

Variable dependent:

- numbikes: indica el nombre de bicicletes que trobem.

Variables explicatives:

- weather_type: variable categòrica que representa el tipus de temps que fa (sol, ennuvolat, pluja suau, intensa, ...)
- updatetime: variable contínua que representa el moment del temps pel qual es vol la predicció
- dayMoment: variable contínua que representa el temps en segons des de l'inici del dia
- weekday: variable categòrica que representa el dia de la setmana

Resultats-Nombre de bicicletes

- El sistema ara prediu el nombre de bicicletes que hi poden haver en un moment del temps.
- Hem considerat un resultat correcte quan:
  - Si la realitat es que hi ha alguna bici: considerem correcte quan el nombre de bicicletes que predim es +- 20% el nombre de bicicletes reals a l'estació.
  - Si la realitat és que no hi ha bicicletes a l'estació, considerem encertada la predicció quan el programa prediu 0 bicicletes.

		LABORABLES			FESTIUS			TOTAL		
Est.	**pBikes**	**Encert**	**Errors**	**%Enc.**	**Encert**	**Errors**	**%Enc.**	**Encert**	**Errors**	**%Enc.**
30	1	1.223	2.111	37%	1.280	1.511	46%	2.503	3.622	40,9%
30	0	0	58	0%	0	44	0%	0	102	0,0%
42	1	560	2.364	19%	261	2.235	10%	821	4.599	15,1%
42	0	130	338	28%	25	314	7%	155	652	19,2%
74	1	348	2.677	12%	260	2.241	10%	608	4.918	11,0%
74	0	0	367	0%	0	334	0%	0	701	0,0%
366	1	429	2.640	14%	33	2.180	1%	462	4.820	8,7%
366	0	0	323	0%	0	622	0%	0	945	0,0%
	1	**2.560**	**9.792**	**21%**	**1.834**	**8.167**	**18%**	**4.394**	**17.959**	**19,7%**
	0	**130**	**1.086**	**11%**	**25**	**1.314**	**2%**	**155**	**2.400**	**6,1%**
		2.690	**10.878**	**20%**	**1.859**	**9.481**	**16%**	**4.549**	**20.359**	**18,26%**

Resultats-Hi ha bicicletes?

- El sistema ara prediu el nombre de bicicletes que hi poden haver en un moment del temps.
- Hem considerat un resultat correcte quan:
  - Si la realitat es que hi ha alguna bici: considerem correcte quan el nombre de bicicletes que prediem és de 1 o més.
  - Si la realitat és que no hi ha bicicletes a l'estació, considerem encertada la predicció quan el programa prediu 0 bicicletes.

		LABORABLES			FESTIUS			TOTAL		
Est.	**pBikes**	**Encert**	**Errors**	**%Enc.**	**Encert**	**Errors**	**%Enc.**	**Encert**	**Errors**	**%Enc.**
30	1	3.334	0	100%	2.791	0	100%	6.125	0	100,0%
30	0	0	58	0%	0	44	0%	0	102	0,0%
42	1	2.924	0	100%	2.496	0	100%	5.420	0	100,0%
42	0	0	468	0%	0	339	0%	0	807	0,0%
74	1	3.025	0	100%	2.501	0	100%	5.526	0	100,0%
74	0	0	367	0%	0	334	0%	0	701	0,0%
366	1	3.069	0	100%	2.213	0	100%	5.282	0	100,0%
366	0	0	323	0%	0	622	0%	0	945	0,0%
	1	**12.352**	**0**	**100%**	**10.001**	**0**	**100%**	**22.353**	**0**	**100,0%**
	0	**0**	**1.216**	**0%**	**0**	**1.339**	**0%**	**0**	**2.555**	**0,0%**
		12.352	**1.216**	**91%**	**10.001**	**1.339**	**88%**	**22.353**	**2.555**	**89,74%**

Comentari

- Encert molt baix en la predicció del nombre de bicicletes, i un encert molt alt en predir que hi ha bicicletes.
- Encert nul en predir que no hi ha bicicletes, el model sempre ha predit que hi ha bicicletes.

Resultats globals

	LABORABLES			FESTIUS			TOTAL		
pBikes	Encert	Errors	%Enc.	Encerts	Errors	%Enc.	Encerts	Errors	%Enc.
V1	v1-Prova bàsica								
1	16.351	593	97%	5.282	32	99%	21.633	625	97,2%
0	252	2.036	11%	19	247	7%	271	2.283	10,6%
	16.603	2.629	86%	5.301	279	95%	21.904	2.908	88,28%
V2	v2-dies de la setmana "dummies"								
1	16.332	707	96%	5.225	89	98%	21.557	796	96,4%
0	154	2.135	7%	0	266	0%	154	2.401	6,0%
	16.486	2.842	85%	5.225	355	94%	21.711	3.197	87,2%
V3	v3-Temps booleà (plou/no plou).								
1	16.560	479	97%	5.174	140	97%	21.734	619	97,2%
0	210	2.079	9%	55	211	21%	265	2.290	10,4%
	16.770	2.558	87%	5.229	351	94%	21.999	2.909	88,32%
V4bis	v4bis- Temps booleà i temperatura								
1	16.392	647	96%	5.240	74	99%	21.632	721	96,8%
0	252	2.037	11%	0	266	0%	252	2.303	9,9%
	16.644	2.684	86%	5.240	340	94%	21.884	3.024	87,9%
V5	v5-Temps categoritzat i Temperatura								
1	16.396	643	96%	5.243	71	99%	21.639	714	96,8%
0	240	2.049	10%	0	266	0%	240	2.315	9,4%
	16.636	2.692	86%	5.243	337	94%	21.879	3.029	87,84%
V6a	v6-Predim el nombre de bicicletes (nombre)								
1	2.560	9.792	21%	1.834	8.167	18%	4.394	17.959	19,7%
0	130	1.086	11%	25	1.314	2%	155	2.400	6,1%
	2.690	10.878	20%	1.859	9.481	16%	4.549	20.359	18,26%
V6b	v6-Predim el nombre de bicicletes								
1	12.352	0	100%	10.001	0	100%	22.353	0	100,0%
0	0	1.216	0%	0	1.339	0%	0	2.555	0,0%
	12.352	1.216	91%	10.001	1.339	88%	22.353	2.555	89,74%

Sembla ser que el resultat més compensat és v3, doncs ens dona la més alta probabilitat d'encerts tant quan hi ha bicicletes com quan no n'hi ha.

5.3 Model De Regressió Random Forest Curt/Llarg

Hem explorat la possibilitat de fer una predicció diferent segons el període de temps en que es demana. La tecnologia que fem servir per Random Forest és "R", en el que hem de crear el model sobre el que farem la predicció. Aquest model es construeix sobre dades existents, això implicaria que cada vegada que un usuari ens demana una predicció idealment el reconstruiríem. Fer això és poc pràctic en el dia a dia, ja que el cost de construir el model és força alt i el temps que es triga a construir-lo també.

Amb els recursos de que disposem no és viable poder fer aquesta reconstrucció del model per cada predicció, **així que sempre partim d'un model calculat la nit anterior**. Això vol dir que si un usuari està demanant conèixer el nombre de bicicletes d'aquí 1 hora, el sistema no té en compte l'ocupació actual de l'estació.

Així doncs, hem pensat en aplicar un sistema lleugerament diferent per les prediccions a curt termini:

- Diferenciar curt/llarg termini
- Per una predicció llarg: fer servir l'algorisme normal (amb el model v3)
- Per una predicció curt:
    - Predir el nombre de bicicletes que hi hauria d'haver ara mateix (amb el model v6)
    - Predir el nombre de bicicletes normalment (amb el model v6)
    - Calcular l'offset de les dues prediccions, és a dir segons el model quantes bicicletes guanyarem o perdrem.
    - Demanar el nombre de bicicletes actuals
    - Aplicar l'offset de predicció sobre el nombre de bicicletes actuals
    - Aquest serà el nombre de bicicletes que informarem com a resultat de la predicció.

Les prediccions s'han fet amb les mateixes variables explicatives que el cas v3, és a dir, aplicant el millor resultat assolit:

Variable dependent:

- pbikes: indica la probabilitat de trobar o no bicicletes en una estació

Variables explicatives:

- weather_type: *variable categòrica booleana que representa si hi ha precipitacions o no, a diferència dels casos anteriors aquí hem volgut simplificar aquesta variable per "ajudar" a l'algorisme de predicció*
- updatetime: variable contínua que representa el moment del temps pel qual es vol la predicció
- dayMoment: variable contínua que representa el temps en segons des de l'inici del dia
- weekday: variable categòrica que representa el dia de la setmana

Model curt termini envers Model v3 per 1 dia

Compararem els resultats obtinguts d'executar la predicció sobre les dades d'un dia, i ho compararem amb el millor model v3 d'un sol dia. És a dir, tractarem d'esbrinar si la predicció a curt termini és millor si està feta amb l'últim nombre de bicicletes disponible abans de la predicció:

Resultats del model a curt termini comparat amb v3

	Hi ha bici			No hi ha bici			Total		
Model	**Encert**	**Errors**	**%Enc.**	**Encert**	**Errors**	**%Enc.**	**Encert**	**Errors**	**%Enc.**
30 min	4.147	374	91,73%	178	345	34,03%	**4.325**	**719**	**85,75%**
60 min	4.135	386	91,46%	163	360	31,17%	**4.298**	**746**	**85,21%**
90 min	4.080	441	90,25%	95	428	18,16%	**4.175**	**869**	**82,77%**
120 min	4.079	442	90,22%	68	455	13,00%	**4.147**	**897**	**82,22%**
v3	4.330	191	95,78%	35	488	6,69%	**4.365**	**679**	**86,54%**

Comentari

- Observem que els resultats totals del model v3 són millors que els dels nous models a curt termini.
- Els encerts pels casos que no hi ha bicicletes són considerablement millors en els models curt-termini que en el model v3.

Model curt i llarg termini (junts)

Així doncs implementarem un algorisme que sigui lleugerament diferent pel curt que pel llarg termini. És a dir, calcularem les prediccions de dues maneres diferents, segons el termini de temps que l'usuari requereixi.

	Hi ha bici			No hi ha bici			Total		
Model	**Encert**	**Errors**	**%Enc.**	**Encert**	**Errors**	**%Enc.**	**Encert**	**Errors**	**%Enc.**
v3no24 + 30 min	21.551	802	96,41%	408	2.147	15,97%	**21.959**	**2.949**	**88,16%**
v3no24 + 60 min	21.539	814	96,36%	393	2.162	15,38%	**21.932**	**2.976**	**88,05%**
v3no24 + 90 min	21.484	869	96,11%	325	2.230	12,72%	**21.809**	**3.099**	**87,56%**
v3no24 + 120 min	21.483	870	96,11%	298	2.257	11,66%	**21.781**	**3.127**	**87,45%**
v3	4.330	191	95,78%	35	488	6,69%	**4.365**	**679**	**86,54%**

Comentari

- Observem que els resultats totals del model v3 són millors que els dels nous models combinats
- **De totes maneres preferim un model combinat, ja que és el més compensat pels dos casos.**

6 Anàlisi

6.1 Anàlisi de temps

Tasca	Perfil	Hores Reals	Hores Planificades	Desviació
Backlog de Producte		**104**	**192**	**-88**
Descripció Sistema	*Cap de Projecte*	8	32	-24
Serveis a l'usuari	*Cap de Projecte*	8	16	-8
Mapa de components	*Cap de Projecte*	32	40	-8
Descripció Motor Predicció	*Cap de Projecte*	8	24	-16
Plataforma App Backend	*Desenvolupador backend*	24	56	-32
Plataforma Base de Dades	*Desenvolupador backend*	24	24	0
Desenvolupament		**480**	**348**	**132**
Captura de Dades	*Desenvolupador backend*	80	48	32
App usuari final	*Desenvolupador App Android*	80	48	32
Motor de predicció	*Desenvolupador backend*	120	140	-20
Proves del Sistema	*Control de qualitat*	80	40	40
Altres				
Documentació	*Cap de Projecte*	120	72	48
Total		**584**	**540**	**44**

Causes Desviació

- Backlog de producte: ha estat negativa, és a dir, he invertit menys temps del que estava previst. Això és així per què no vaig fer una documentació detallada dels requisits del sistema, ni vaig fer una descripció dels components del sistema. Per altra banda la plataforma backend va estar molt senzilla de posar en marxa amb la tecnologia de containers que vaig emprar.
- Desenvolupament: ha estat una desviació significativa, motivat per una definició poc profunda dels requeriments, que m'ha exigit més esforç a l'hora de programar. Les proves del sistema s'han allargat per contemplar diferents casos amb els quals m'he anat trobant.

6.2 Anàlisi financer

El cost econòmic d'aquest projecte es divideix en:

- Recursos Humans
- Infraestructura

En altres projectes podríem tenir en compte el cost del software però en aquest cas no és necessari, ja que tot el software utilitzat és lliure i/o gratuït.

Com a cost de recursos humans ho calcularem a partir de les hores dedicades al projecte donant un valor de mercat als diferents rols.

Recursos Humans

Recurs	Sou Anual [1]	Cost Anual [2]	Cost Horari	Hores	Cost Projecte
Cap de Projecte	39.375,00 €	59.062,50 €	32,81 €	176	5.775,00 €
Desenvolupador backend	30.000,00 €	45.000,00 €	25,00 €	248	6.200,00 €
Desenvolupador App Android	33.500,00 €	50.250,00 €	27,92 €	80	2.233,33 €
Control de qualitat	29.000,00 €	43.500,00 €	24,17 €	80	1.933,33 €
Cost Recursos Humans Projecte				**584**	**16.141,67 €**

Infraestructura

Infraestructura	Des de	Fins a	Mesos	Cost Mensual	Cost Total
Servidor 1 GB de Ram i 1 CPU	Gener	Juliol	7	5,00 €	35,00 €
Actualització 4GB de Ram i 2 CPUs	Agost	Setembre	2	40,00 €	80,00 €
Cost Infraestructura					**115,00 €**

Cost Total

Concepte	Import
Recursos Humans	16.141,67 €
Infraestructura	115,00 €
Cost Total Projecte	**16.256,67 €**

[1] http://www.techsalarycalculator.com/
[2] Costos socials i empresarials del 50% del sou.

6.3 Anàlisi de competències

En el treball he tractat de posar en valor pràctic tot un seguit de matèries, entre les que voldria destacar:

- **Enginyeria de software**: per tal de construir i dissenyar el sistema, és a dir, que tractarem de seguir pautes i documentació segons un criteri d'enginyeria.
    - Planificació del projecte
    - Anàlisi financer de costos
    - Metodologia

- **Programació**: la base del projecte és un programari, per tant posarem en valor coneixements i habilitats adquirides de programació.
    - Desenvolupament del frontend
    - Desenvolupament del backend

- **Bases de Dades**: el motor de predicció es basa en dades acumulades, per tant haurem de dissenyar un esquema on quedin emmagatzemats les dades en format cru.
    - Coneixements de base de dades-administració
    - Coneixements de MySQL

- **Matemàtiques i estadística**: l'algoritme d'estimació tindrà com a base una predicció estadística basada en diferents dades aplicant diferents pesos.
    - Coneixements d'estadística que m'han permès iniciar-me en la predicció mitjançant el model Random Forest

7 Conclusions

7.1 Comparativa amb la competència

He comparat els resultats de la meva estimació amb www.bicintime.com, l'únic competidor que dóna servei de predicció. Els resultats són els següents:

Estació	Dia	Mes	Any	Hora	Min	Bicicletes Reals	Bicing Stats	UPF
42	24	8	2016	12	24	7	9	8
42	24	8	2016	12	8	6	10	8
30	24	8	2016	12	4	14	12	3
30	24	8	2016	12	34	14	15	4
30	24	8	2016	12	35	14	14	5
30	24	8	2016	12	36	14	15	6
366	24	8	2016	12	36	22	21	6
366	24	8	2016	12	43	22	21	6
366	24	8	2016	12	45	22	22	6
74	24	8	2016	12	49	1	1	6
74	24	8	2016	12	50	1	1	4
74	24	8	2016	12	51	1	0	2
74	24	8	2016	12	52	1	0	1
74	24	8	2016	12	53	1	0	1

Observem que:

- La meva predicció, encara que a vegades no és la més acurada, és la que menys dispersió ofereix en conjunt.
    - La competència ha estimat en algunes ocasions uns valors de bicicletes molt diferents de la realitat (6 bicicletes, quan hi ha 15, etc...)
- La meva predicció és la més ajustada en més ocasions (9 guanyo jo, 1 empat, i només en 4 ocasions la competència és més ajustada).
- La meva predicció té 3 casos en els quals prediu que no hi haurà bicicletes, i en canvi hi ha 1 bicicleta.

Conclusió:

Tot i que les dades d'aquest sistema de les que disposo són poques, ja que les hem hagut de recollir manualment, els resultats semblen indicar que el meu sistema és més precís que el seu. Més endavant m'agradaria fer un sistema de recol·lecció de dades de la seva API per tal de poder fer una comparativa rigorosa.

7.2 El futur de Bicing Stats

M'agradaria continuar millorant l'aplicació i el sistema construït per tal que pugui ser una aplicació comercial amb les següents millores:

Millorar la qualitat de la predicció:

Fent servir Cross-Validation de dades per mirar "d'ajudar" a l'algorisme de predicció, ja que tot i tenir un percentatge d'encerts molt alt quan prediu que hi ha bicicletes, el percentatge d'encerts quan prediu que no n'hi haurà és molt baix actualment.

Voldria provar d'afegir informació d'esdeveniments especials a la ciutat que pugui afectar a l'ús de Bicing (partits de futbol, concerts, curses populars, ...), ja que la meva experiència personal em fa pensar que tenen una forta incidència en l'ocupació de les estacions però no tenen correlació amb cap de les dades que utilitzem per fer les prediccions actuals, el que ho fa una dada rellevant.

A més, també seria molt interessant afegir un sistema automatitzat d'informes de la qualitat de les estimacions, per tal de ser utilitzada per millorar la qualitat de l'algoritme.

Optimitzar el rendiment del sistema:

Actualment el temps de resposta percebut per l'usuari és correcte, però el càlcul del model resulta molt lent (és nocturn), sobretot pel fet que és una consulta relativament complicada a la base de dades, que podríem millorar si fos desnormalitzada.

Calculo el model de predicció utilitzant R, però podria ser interessant fer comparatives amb implementacions de l'algoritme en altres llenguatges (o fins i tot fer-ne una implementació pròpia).

Afegir més plataformes:

És necessari tenir tant una versió web de l'aplicació com una versió iOS, ja que són plataformes amb molt de pes i m'assegurarien poder arribar al màxim de públic possible.

Una iniciativa que m'agradaria valorar seria crear una aplicació per Smartwatch, ja que quan som a la bicicleta podem voler realitzar alguna consulta i evitar treure el mòbil pot ser un gran valor afegit pels usuaris.

Respecte a la informació que es dóna a l'usuari:

Voldria informar a l'usuari del nombre de places disponibles a l'estació. Els criteris que hi aplicaré seran els mateixos que es fan servir a la predicció d'ocupació de l'estació.

Afegir una opció per fer consultes de com anar d'un lloc a un altre, de tal manera que l'aplicació proporcionaria a l'usuari la probabilitat de trobar bicicletes a l'estació de sortida, el temps de trajecte i la probabilitat de trobar espais buits a l'arribada (similar a com ho fa Bicintime, el meu competidor).

Afegir un històric de les consultes que ha fet l'usuari amb l'opció de marcar certes estacions o trajectes com preferides, de tal manera que facilitaria l'ús de l'estació a l'usuari.

També hauria de considerar afegir una eina per obtenir feedback de possibles problemes de l'aplicació, ja sigui demanant-ho a l'usuari directament o instrumentant l'aplicació per tal d'obtenir dades d'ús de memòria, CPU, ...

Per últim, i pensant en tenir una aplicació que com a mínim no suposi un cost de manteniment, s'hauria d'optar bé per demanar donacions voluntàries als usuaris o per introduir alguna característica de monetització, ja sigui mostrar anuncis, fer l'aplicació de pagament o fer que alguna característica només estigui disponible per usuaris pagadors (com per exemple l'historial de consultes).

7.3 Opinió personal

Aquest projecte ha estat molt interessant per mi, ja que m'ha servit per adquirir diversos coneixements nous.

Per començar he tingut l'oportunitat d'emprar Go, que és molt orientat a servidor i que m'ha semblat molt més eficaç que PHP, el llenguatge que faig servir habitualment al meu dia a dia a la feina. Go té molt bon rendiment i pel fet de tenir tipus estàtics proporciona poques sorpreses. A més, és un llenguatge que em recorda força a una versió simplificada de C++, que va ser el meu primer llenguatge de programació.

També he pogut aprendre R, del que en coneixia només el "hello world" i que m'ha semblat molt fàcil d'utilitzar i sorprenentment eficient, tenint en compte que es tracta d'un llenguatge interpretat. R m'ha facilitat poder iniciar-me a la "data science" amb aquest projecte, un camp en el qual m'agradaria aprofundir per tal de ser capaç de millorar amb fonament el sistema de predicció de l'aplicació. Aquests coneixements em facilitaran en gran mesura futures tasques d'anàlisi de dades les quals són molt útils en tot tipus de projectes (ja sigui per descobrir què funciona i què no o per tal de realitzar prediccions, igual que he fet a aquest projecte).

La meva experiència amb Docker, m'ha semblat molt constructiva. Ja havia utilitzat contenidors de Docker a la feina, però havia construït l'estructura des de zero. Tot i que la posada en marxa del sistema de contenidors em va consumir força temps en un moment inicial, crec que a la llarga va ser una bona inversió, ja que em va evitar molts problemes potencials causats per possibles diferències de versions entre el servidor i el client. Amb el que sé ara no dubtaria en tornar a utilitzar Docker en el meu proper projecte.

8 Bibliografia

Documentació oficial de Go

https://golang.org/

Documentació oficial d'Android

https://developer.android.com/

Documentació oficial de R

https://cran.r-project.org/

Documentació oficial de Docker

https://docs.docker.com/

Resolució de dubtes puntuals durant el desenvolupament

http://www.stackoverflow.com

Informació sobre Random Forest

https://en.wikipedia.org/wiki/Random_forest

https://www.youtube.com/watch?v=loNcrMjYh64ç

https://www.quora.com/What-are-the-advantages-of-different-classification-algorithms

Intel·ligència artificial i Aprenentatge automàtic

www.udacity.com

Patrons d'aplicació mòbil:

http://appdesignbook.com/es/contenidos/presentacion/

Concepte d'arquitectura

http://www.iainstitute.org/what-is-ia

Elements que configuren l'arquitectura d'una aplicació

https://es.wikipedia.org/wiki/Arquitectura_de_la_informaci%C3%B3n

Metodologia Scrum

http://openaccess.uoc.edu/webapps/o2/bitstream/10609/45670/7/jtiernobTFC0116mem%C3%B2ria.pdf